Please type a plus sign (+) inside this box [+] 06-26-00 PTO/SB/05 (2/98)

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for nonprovisional applications under 37 CFR § 1.53(b))

Attorney Docket No.	500689.01
First Inventor or Application Identifier	James R. Peterson
Title	APPARATUS AND METHOD FOR DYNAMICALLY DISABLING FAULTY EMBEDDED MEMORY IN A GRAPHIC PROCESSING SYSTEM
Express Mail Label No.	EL476401367US

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

ADDRESS TO: Box Patent Application, Assistant Commissioner for Patents, Washington, D.C. 20231

1. [X] General Authorization Form & Fee Transmittal *(Submit an original and a duplicate for fee processing)*
2. [X] Specification [Total Pages] 22 *(preferred arrangement set forth below)*
   - Descriptive Title of the Invention
   - Cross References to Related Applications
   - Statement Regarding Fed sponsored R & D
   - Reference to Microfiche Appendix
   - Background of the Invention
   - Brief Summary of the Invention
   - Brief Description of the Drawings *(if filed)*
   - Detailed Description
   - Claim(s)
   - Abstract of the Disclosure
3. [X] Drawing(s) *(35 USC 113)* [Total Sheets] 5
4. Oath or Declaration [Total Pages] 2
   a. [X] Newly executed (original or copy)
   b. [ ] Copy from a prior application (37 CFR 1.63(d)) *(for continuation/divisional with Box 17 completed)*
      i. [ ] DELETION OF INVENTOR(S) Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b)
5. [ ] Incorporation By Reference *(useable if box 4b is checked)* The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Box 4b, is considered to be part of the disclosure of the accompanying application and is hereby incorporated by reference therein.
6. [ ] Microfiche Computer Program *(Appendix)*
7. Nucleotide and Amino Acid Sequence Submission *(if applicable, all necessary)*
   a. [ ] Computer-Readable Copy
   b. [ ] Paper Copy (identical to computer copy)
   c. [ ] Statement verifying identity of above copies

ACCOMPANYING APPLICATION PARTS

8. [X] Assignment Papers (cover sheet & document(s))
9. [X] 37 CFR 3.73(b) Statement *(when there is an assignee)* [X] Power of Attorney
10. [ ] English Translation Document *(if applicable)*
11. [ ] Information Disclosure Statement (IDS)/PTO-1449 [ ] Copies of IDS Citations
12. [ ] Preliminary Amendment
13. [X] Return Receipt Postcard
14. [ ] Small Entity Statement(s) [ ] Statement filed in prior application, Status still proper and desired
15. [ ] Certified Copy of Priority Document(s) *(if foreign priority is claimed)*
16. [X] Other: Certificate of Express Mail Check

17. If a **CONTINUING APPLICATION,** *check appropriate box and supply the requisite information below and in a preliminary amendment*

[ ] Continuation [ ] Divisional [ ] Continuation-In-Part (CIP) of prior Application No.: _____ Filed _____

Prior application information: Examiner _____ Group / Art Unit _____

[ ] Claims the benefit of Provisional Application No. _____

CORRESPONDENCE ADDRESS

Kimton N. Eng, Esq.
Dorsey & Whitney LLP
1420 Fifth Avenue, Suite 3400
Seattle, Washington 98101-4010
(206) 903-8800 *phone*
(206) 903-8820 *fax*

Respectfully submitted,

TYPED or PRINTED NAME Kimton N. Eng REGISTRATION NO. 43,605

SIGNATURE _____ Date June 23 2000

Rendition\500689.01\sb-05 500689.01.doc

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants	: James R. Peterson and William Radke	Attorney Docket No.: 500689.01
Filed	: June 23, 2000	
Title	: APPARATUS AND METHOD FOR DYNAMICALLY DISABLING FAULTY EMBEDDED MEMORY IN A GRAPHIC PROCESSING SYSTEM	

CERTIFICATE OF MAILING BY "EXPRESS MAIL"

Box Patent Application
Assistant Commissioner of Patents
Washington, D.C. 20231

Sir:

I hereby certify that the enclosures listed below are being deposited with the United States Postal Service "EXPRESS MAIL Post Office to Addressee" service under 37 C.F.R. § 1.10, Mailing Label Certificate No. EL476401367US, on June 23, 2000, addressed to Box Patent Application, Assistant Commissioner for Patents, Washington, DC 20231.

Respectfully submitted,

DORSEY & WHITNEY LLP

Susan Johnson
/ for West Courier

Enclosures:
- Postcard
- Check
- Form PTO/SB/05
- General Authorization Under 37 C.F.R. § 1.136(a)(3) and Fee Transmittal (+ copy)
- Specification, Claims, Abstract (22 pages)
- 5 Sheets of Drawings (Figures 1-5)
- Declaration
- Election and Power of Attorney
- Form PTO-1595
- Assignment

o:\ip\documents\clients\rendition\500689.01\500689 01 cert of mlg doc

APPARATUS AND METHOD FOR DYNAMICALLY DISABLING FAULTY EMBEDDED MEMORY IN A GRAPHIC PROCESSING SYSTEM

TECHNICAL FIELD

The present invention is related generally to the field of computer graphics, and more particularly, to a memory system for use in a computer graphics processing system.

BACKGROUND OF THE INVENTION

Graphics processing systems are included in computer systems to assist a host microprocessor with some of the processing burden involved with rendering graphic images. Graphics processing systems include semiconductor devices having the entire system fabricated on a single semiconductor substrate, as well as peripheral component cards inserted into a computer system and having multiple devices located on a single component card. As computer applications have become more computer graphics intensive, the need for graphics processing systems having the ability to process graphics data and render graphics images at high speeds has also increased. For example, high resolution three-dimensional (3D) computer animation, which at one time was limited to computer workstations, has become standard in the computer gaming industry for home personal computers. Consequently, current graphics processing systems must be capable of processing the 3D graphics data and rendering frames of graphics images fast enough to avoid image stuttering while maintaining image quality.

To this end, graphics processing systems are typically designed with high-speed graphics processors and a limited amount memory fabricated on the same semiconductor substrate as the graphics processor. Such memory has low-access times and is commonly referred to as embedded memory. The use of embedded memory in a graphics processing system has the advantage of providing a limited amount of high-speed memory to facilitate fast graphics processing. The low access times of the embedded memory are in part the result of the physical proximity of the embedded

memory to the graphics processor, as well as the fact that the embedded memory is dedicated for the purpose of graphics processing.

As semiconductor fabrication techniques have improved, and memory cell design has become more sophisticated, the amount of embedded memory available in a graphics processing system has dramatically increased. It is now possible to find a graphics processing system that includes as much as 12 MBytes of embedded memory. However, a consequence of increasing the available embedded memory is that there is now an increased chance for the embedded memory to have defective memory locations. As a result of the embedded memory being located on the same semiconductor substrate as the graphics processor, a defective memory location in the embedded memory would render an otherwise functional graphics processing processor unusable.

Consequently, as with most memory arrays, the embedded memory of a graphics processing system is designed to include redundant memory. The redundant memory is substituted for the defective embedded memory, and thus, increases the production yield of fully functional graphics processing systems. The redundant memory is substituted for the defective memory by re-mapping memory addresses of the faulty memory to the functional redundant memory. The process of testing for faulty memory, and replacing it with redundant memory occurs as part of the manufacturing process of the graphics processing system. The re-mapping of memory addresses occurs internally to the graphics processing system and appears transparent to the computer system.

Although redundant memory has been effective in increasing the production yield of graphics processing devices, as the amount of embedded memory has increased, so has the amount of redundant memory in order to maintain the relative recovery rate from substituting faulty embedded memory with available redundant memory. As a result, more area on the semiconductor substrate is occupied by redundant memory. Where reducing the size of the graphics processing device is desired, allocating space on the semiconductor substrate to redundant memory, that is,

memory that may or may not be used depending on the amount of faulty embedded memory, is not likely to be a preferable alternative.

Therefore, there is a need for a graphics processing system and method where an otherwise functional graphics processing system having more faulty memory than can be substituted by available redundant memory can still be used for graphics processing.

SUMMARY OF THE INVENTION

The invention is directed to a memory subsystem and method for ignoring faulty memory sub-arrays and assigning functional memory sub-arrays to accessible memory blocks. The memory subsystem includes a memory array segmented into a plurality of memory sub-arrays and a memory controller that accesses the memory array. The memory controller references a register storing a pointer value that assigns a memory block to each of the functional memory sub-arrays of the memory array. The faulty memory sub-arrays are left unassigned so that they are ignored by the memory controller. Upon receiving a memory access request to a particular memory block, the memory controller accesses the functional memory sub-array assigned to the particular memory block by the pointer value.

The memory subsystem may further include a second memory array and a second memory controller coupled to the first memory controller through a memory controller bus. The functional memory sub-arrays of the second memory array are also assigned to memory blocks. A respective start address and size value define the addressable memory area of each of the memory arrays. Each memory controller references the respective start address and size value upon receiving a memory access request to determine whether the particular memory block is assigned to a sub-array within the memory array to which it is coupled. The memory access request is passed to the other memory controller over the memory controller bus if it is determined that the memory block is not within the addressable memory area.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of a computer system in which embodiments of the present invention are implemented.

Figure 2 is a block diagram of a graphics processing system in the computer system of Figure 1.

Figure 3 is a block diagram representing a memory system according to an embodiment of the present invention.

Figure 4 is a block diagram of a memory system having a distributed memory controller arrangement.

Figure 5 is a block diagram representing the memory system of Figure 4 according to another embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

Embodiments of the present invention provide a memory system that allows an otherwise functional graphics processing system that has faulty blocks of memory to be used for graphics processing. The faulty blocks of memory are ignored during memory access by programming and storing appropriate values in a register, or registers, that are consulted by a memory controller during memory access.

Certain details are set forth below to provide a sufficient understanding of the invention. However, it will be clear to one skilled in the art that the invention may be practiced without these particular details. In other instances, well-known circuits, control signals, timing protocols, and software operations have not been shown in detail in order to avoid unnecessarily obscuring the invention.

Figure 1 illustrates a computer system 100 in which embodiments of the present invention are implemented. The computer system 100 includes a processor 104 coupled to a host memory 108 through a memory/bus interface 112. The memory/bus interface 112 is coupled to an expansion bus 116, such as an industry standard architecture (ISA) bus or a peripheral component interconnect (PCI) bus. The computer system 100 also includes one or more input devices 120, such as a keypad or a mouse, coupled to the processor 104 through the expansion bus 116 and the memory/bus

interface 112. The input devices 120 allow an operator or an electronic device to input data to the computer system 100. One or more output devices 120 are coupled to the processor 104 to provide output data generated by the processor 104. The output devices 124 are coupled to the processor 104 through the expansion bus 116 and memory/bus interface 112. Examples of output devices 124 include printers and a sound card driving audio speakers. One or more data storage devices 128 are coupled to the processor 104 through the memory/bus interface 112 and the expansion bus 116 to store data in, or retrieve data from, storage media (not shown). Examples of storage devices 128 and storage media include fixed disk drives, floppy disk drives, tape cassettes and compact-disc read-only memory drives.

The computer system 100 further includes a graphics processing system 132 coupled to the processor 104 through the expansion bus 116 and memory/bus interface 112. Optionally, the graphics processing system 132 may be coupled to the processor 104 and the host memory 108 through other types of architectures. For example, the graphics processing system 132 may be coupled through the memory/bus interface 112 and a high speed bus 136, such as an accelerated graphics port (AGP), to provide the graphics processing system 132 with direct memory access (DMA) to the host memory 108. That is, the high speed bus 136 and memory bus interface 112 allow the graphics processing system 132 to read and write host memory 108 without the intervention of the processor 104. Thus, data may be transferred to, and from, the host memory 108 at transfer rates much greater than over the expansion bus 116. A display 140 is coupled to the graphics processing system 132 to display graphics images. The display 140 may be any type of display, such as a cathode ray tube (CRT), a field emission display (FED), a liquid crystal display (LCD), or the like, which are commonly used for desktop computers, portable computers, and workstation or server applications.

Figure 2 illustrates circuitry included within the graphics processing system 132 for performing various three-dimensional (3D) graphics functions. As shown in Figure 2, a bus interface 200 couples the graphics processing system 132 to the expansion bus 116. In the case where the graphics processing system 132 is coupled to the processor 104 and the host memory 108 through the high speed data bus 136 and

the memory/bus interface 112, the bus interface 200 will include a DMA controller (not shown) to coordinate transfer of data to and from the host memory 108 and the processor 104. A graphics processor 204 is coupled to the bus interface 200 and is designed to perform various graphics and video processing functions, such as, but not limited to, generating vertex data and performing vertex transformations for polygon graphics primitives that are used to model 3D objects. The graphics processor 204 is coupled to a triangle engine 208 that includes circuitry for performing various graphics functions, such as clipping, attribute transformations, rendering of graphics primitives, and generating texture coordinates for a texture map. A pixel engine 212 is coupled to receive the graphics data generated by the triangle engine 208. The pixel engine 212 contains circuitry for performing various graphics functions, such as, but not limited to, texture application or mapping, bilinear filtering, fog, blending, and color space conversion.

A memory controller 216 coupled to the pixel engine 212 and the graphics processor 204 handles memory requests to and from an embedded memory 220. The embedded memory 220 stores graphics data, such as source pixel color values and destination pixel color values. The memory controller 216 includes various registers that, as will be explained in more detail below, store values that are used to disable faulty blocks of memory during the access of the embedded memory 220. In this way, a graphics processing system that is otherwise functional, but has a faulty block of embedded memory that cannot be entirely replaced by available redundant memory, can nevertheless still be used for graphics processing.

A display controller 224 coupled to the embedded memory 220 and to a first-in first-out (FIFO) buffer 228 controls the transfer of destination color values to the FIFO 228. Destination color values stored in the FIFO 336 are provided to a display driver 232 that includes circuitry to provide digital color signals, or convert digital color signals to red, green, and blue analog color signals, to drive the display 140 (Figure 1).

Illustrated in Figure 3 is a representation 300 of the memory controller 216 and the embedded memory 220 (Figure 2). The memory controller 216 includes register 304 having fields 312, 316, 320, 324, and 330, and the embedded memory 220

includes an embedded memory array 308 segmented into four memory blocks 334, 338, 342, and 346. The register 304 is described above as being included in the memory controller 216, however, the register 304 is not limited to this location, and may be located externally to the memory controller. The register 304 includes a memory valid field 312 for storing a value indicative of which memory blocks 334, 338, 342, and 346, are functional, and further includes four BANK fields, 316, 320, 324, and 330. Each field corresponds to one of the four blocks of memory 334, 338, 342, and 346. Although the embedded memory array 308 has been illustrated in Figure 3 as being segmented into four memory blocks, it will be appreciated that the embedded memory array 308 may be segmented into greater or fewer memory blocks. Moreover, the particular size of each memory block into which the embedded memory 308 is segmented is a detail that may be changed without departing from the scope of the present invention. As such, the number of memory blocks into which the embedded memory array 308 is segmented, or the size of each of the memory blocks, should not limit the scope of the present invention.

Figure 3a illustrates a situation where it has been determined from functional testing that all four blocks of memory 334, 338, 342, and 346 are functional. The value 1111 is written to and stored in the memory valid field 312 to indicate that all four of the blocks of memory may be used. Next, the values indicating which blocks of memory should be accessed when a memory access request is made to a respective memory bank is programmed and stored in the BANK fields 316, 320, 324, 330. The value 00 is stored for the BANK0 field, indicating that when access is requested to a memory location having a memory address in BANK0, memory block 334 will be accessed by the memory controller 216 (Figure 2). The value 01 is stored for the BANK1 field, indicating that when access to a memory location having a memory address in BANK1 is requested, the memory block 338 will be accessed. The values 10 and 11 are programmed and stored for the fields 324 and 330, respectively, so that the appropriate memory block will be accessed as well.

Figure 3b illustrates a case where the memory block 338 has been determined during function testing to be faulty. The values in the fields of the register

304 are programmed so that the memory block 338 is ignored during memory access. That is, the value 1011 is written and stored in the memory valid field 312 to indicate that the second memory block, that is, the memory block 338, is not valid. The values 00, 10, and 11 are programmed for the BANK fields 316, 320, and 324, respectively. The value stored in the BANK3 field is 330 does not matter since the values in the memory valid field 312, and the BANK fields 316, 320, and 324 will prevent access to the fourth bank of memory, namely BANK3. The value 00 in the BANK1 field 316 indicates to the memory controller 216 that access to a memory location having an address in BANK0 will be made to the memory block 334. The value 10 in the BANK2 field 320 indicates that access to a memory location having a memory address in BANK1 will be made to the memory block 342. The value 11 in the BANK2 field 324 indicates that access to a memory location having a memory address in BANK2 will be made to the memory block 346. The memory block 338, which is faulty in the present example, is consequently ignored.

The present example illustrates that although an entire block of memory may be faulty, and cannot be repaired through the use of redundant memory, the graphics processing system 132 with reduced overall available embedded memory may nevertheless still be used for graphics processing since the faulty block of memory is ignored by the memory controller 216 during memory access.

Figure 3c illustrates the case where memory blocks 342 and 346 have been determined during function testing to be faulty. The value 1100 is programmed and stored in the memory valid field 312 to indicate to the memory controller 216 that only memory blocks 334 and 338 are valid. The values 00 and 01 are written to the BANK fields 316 and 320, respectively, so that memory access requests to BANK0 and BANK1 are directed to the correct memory blocks, namely, memory blocks 334 and 338. The value stored in the BANK fields 324 and 330 are immaterial since the memory blocks 342 and 346 are ignored by the memory controller 216. Although the overall available memory of the graphics processing system 132 as represented in Figure 3c is reduced by two blocks of memory, the graphics processing system 132 may still be used for graphics processing.

Figure 4 illustrates a distributed memory controller memory subsystem 400 that may be substituted into a graphics processing system. A more detailed description of a similar memory subsystem is provided in Patent Application No. _____, filed _____, which is incorporated herein by reference. To summarize, the memory subsystem 400 includes two memory controllers 402 and 422 coupled together through a memory controller bus 216. The memory controller bus 216 allows memory access requests, as well as data, to be passed between the two memory controllers 402 and 422. Each of the memory controllers 402 and 422 is coupled to an addressable memory area 412 and 432, respectively, that is defined by two values. The two values are stored in registers 404a-b and 406a-b. Registers 404a and 406a of the memory controller 402 store the start address and memory size for the addressable memory area 412, and registers 404b and 406b of the memory controller 422 store the start address and memory size value for the addressable memory area 432. These values are referenced by the respective memory controller to determine whether a memory access request is to a memory location in the addressable memory area to which the memory controller is coupled.

For example, the arrangement of the memory subsystem 400 allows a memory access request made to the memory controller 402 over request lines 408 to be passed to the memory controller 422 when the requested memory location has a memory address located in the addressable memory area 432. As mentioned previously, the memory controller receiving the memory access request can determine whether the requested address is located within the addressable memory area to which it is coupled by checking the values of the start address and memory size. In the present example, the memory controller 422 receives the memory access request from the memory controller 402, and accesses the addressable memory area 432 to service the memory access request. If the memory access request received by the memory controller 402 is a read command, the memory controller 422 reads the data from the requested memory location and provides the data back to the memory controller that originally received the memory access request, namely, the memory controller 402. If the memory access request was a write command, data to be written to the memory location accompanies

the memory access request that is passed from the memory controller that originally received the memory access request.

Figure 5 illustrates a memory subsystem 500 according to another embodiment of the present invention. The memory subsystem 500 has a distributed memory controller arrangement similar to that illustrated in Figure 4 represented by registers 504 and 554. Each of the registers 504 and 554 are located in a respective memory controller. The memory subsystem 580 further includes addressable memory areas represented by memory arrays 508 and 558. The memory array 508 is segmented into m memory blocks and the memory array 558 is segmented into n memory blocks. The register 504 includes start address field 512, memory size field 514, and memory valid field 516, and a number of BANK fields 518-526. Each of the BANK fields 518-526 corresponds to a memory block in the memory array 508. The register 554 includes start address field 562, memory size field 564, and memory valid field 566, and BANK registers 568-574. Each of the BANK fields 568-574 corresponds to a memory block in the memory array 558. The start address field 512 and memory size field 514 are programmed with the start address and memory size for the memory array 508. As explained previously, these values are referenced by the respective memory controllers in order to determine whether to pass a memory access request it receives to another memory controller in order to access the requested memory location.

The memory valid fields 516 and 566 store a value indicative of the functional memory blocks of the respective memory array. The BANK fields 518-526 and 568-574 store values indicating which blocks of memory will be accessed when a memory access request is made to a respective memory bank. As explained with respect to Figure 3, faulty memory blocks of the memory arrays 508 and 558 may be ignored by the memory controllers 504 and 554 by programming the appropriate values in the memory valid fields 516 and 566, and the BANK fields 518-526 and 568-574.

Although not included in the representation 300 illustrated in Figure 3, the start address and memory size fields of the memory controller 504 and 554 (Figure 5) should be programmed with the appropriate values so that faulty memory blocks may be ignored when the memory arrays 508 and 558 are accessed. For

example, if the memory addresses of the memory arrays 508 and 558 are to appear contiguous, the start address programmed into field 562 should be the next consecutive memory address following the last addressable memory location of the memory array 508. The last addressable memory location can be determined by subtracting one from the sum of the value stored in the start address field 512 and the value stored in the memory size field 514. Where all of the memory blocks of the memory array 508 are functional, the value stored in the memory size field should be equal to the quantity (m x size of a memory block). Consequently, for the memory addresses of the memory arrays 508 and 558 to appear as a contiguous memory space, the value programmed into the start address field 562 should be [start address + (m x size of a memory block)]. However, if two memory blocks of the memory array 508 are faulty, then the value in stored in the memory size field 514 should be adjusted accordingly, that is, the memory size value should be programmed to equal [(m-2) x size of a memory block]. For the memory of memory arrays 508 and 558 to appear contiguous, the value programmed into the start address field 562 should be [start address + ((m-2) x size of a memory block)]. The values stored in the memory valid field 516 and the BANK fields 518 should also reflect the two faulty memory blocks in order for the faulty memory blocks to be ignored when the memory array 508 is accessed.

The previous example illustrates how the values programmed into the registers 504 and 554 of the memory controllers 402 and 422 (Figure 4) allow an otherwise functional graphics processing system having faulty blocks of embedded memory to nevertheless process graphics data. Moreover, the memory subsystem 500 provides flexibility as to how functional memory may be allocated. The functional memory blocks of the memory arrays 508 and 558 may be allocated in a contiguous fashion, as described in the previous example, or in another desired manner based on the graphics application executing on the computer system 100 (Figure 1).

From the foregoing it will be appreciated that, although specific embodiments of the invention have been described herein for purposes of illustration, various modifications may be made without deviating from the spirit and scope of the invention. Accordingly, the invention is not limited except as by the appended claims.

CLAIMS

1. A memory subsystem, comprising:

a first memory array segmented into a plurality of memory sub-arrays, a number of which are functional, each of the functional memory sub-arrays being assigned to a respective block of memory and any faulty memory sub-arrays being left unassigned; and

a memory controller coupled to receive memory access requests to a block of memory to which a functional memory sub-array from the first memory array is assigned and further coupled to the first memory array to access the functional memory sub-array assigned to the requested block of memory.

2. The memory subsystem of claim 1 wherein the first memory array comprises an embedded memory array.

3. The memory subsystem of claim 2 wherein the first memory controller includes a register to store a pointer value for each of the functional memory sub-arrays indicative of the block of memory to which a respective functional memory sub-array is assigned.

4. The memory subsystem of claim 1 further comprising:

a second memory array segmented into a plurality of memory sub-arrays, a number of which are functional, each of the functional memory sub-arrays of the second memory array assigned to a respective block of memory and any faulty memory sub-arrays left unassigned;

a second memory controller coupled to receive memory access requests to a block of memory to which a functional memory sub-array of the second memory array is assigned and further coupled to the second memory array to access the functional memory sub-array assigned to the requested block of memory; and

a memory bus controller coupled between the first and second memory controllers to pass a memory access request from one memory controller to the other in

response to receiving a memory access request to access a memory location within the memory array coupled to the other memory controller.

5. The memory subsystem of claim 2 wherein the first and second memory controllers store a value indicative of the number of functional sub-arrays in the first memory array a second memory array, respectively.

6. The memory subsystem of claim 2 wherein the first and second memory controllers further store a start and size value for the first and second memory array, respectively, the start and size values defining the addressable memory area of the respective memory array.

7. The memory subsystem of claim 6 wherein the start value stored by the second memory controller is the sum of the start value and the size value stored by the first memory controller.

8. A memory subsystem receiving memory access requests, the memory subsystem comprising:

a first memory array segmented into a plurality of memory sub-arrays, a number of which are functional;

a first register to store pointer values directing access to each functional sub-array; and

a first memory controller coupled to the first memory array and the first register to consult the pointer values and determine which functional memory sub-arrays to access in response to receiving the memory access requests.

9. The memory subsystem of claim 8 wherein the first memory array comprises an embedded memory.

10. The memory subsystem of claim 8, further comprising:

a second memory array segmented into a plurality of memory sub-arrays, a number of which are functional;

a second register to store second pointer values directing access to each functional sub-array of the second memory array;

a second memory controller coupled to the second memory array and the second register to consult the pointer values and determine which of the memory sub-arrays of the second memory array to access in response to receiving the memory access requests; and

a memory controller bus coupled between the first and second memory controllers to pass the memory access request to the other memory controller when the memory access request is to a memory location in the other memory array.

11. The memory subsystem of claim 10 wherein the first and second registers further store a value indicative of the number of functional sub-arrays in the first memory array a second memory array, respectively.

12. The memory subsystem of claim 10 wherein the first and second registers further store a start and size value for the first and second memory array, respectively, the start and size values defining the addressable memory area of the respective memory arrays.

13. The memory subsystem of claim 12 wherein the start value stored by the second register is the sum of the start value and the size value stored by the first register.

14. A memory subsystem, comprising:

a memory array segmented into a plurality of memory sub-arrays; and

a memory controller coupled to access the memory array and having a register including a plurality of data fields, the data fields storing a pointer value indicative of which

memory sub-arrays are functional and which memory sub-arrays to access in response to the memory controller receiving a memory access request.

15. The memory subsystem of claim 14 wherein the memory array comprises an embedded memory array fabricated on a semiconductor substrate with the memory controller.

16. The memory subsystem of claim 14, further comprising:

a second memory array segmented into a plurality of memory sub-arrays;

a second memory controller coupled to access the second memory array and having a register including a plurality of data fields, the data fields of the second memory controller storing a pointer value indicative of which memory sub-arrays of the second memory array are functional and which to access in response to the second memory controller receiving a memory access request; and

a memory controller bus coupled between the memory controller and the second memory controller on which the memory access request may be passed from one memory controller to the other.

17. The memory subsystem of claim 16 wherein the second memory array is an embedded memory fabricated on the same semiconductor substrate as the memory array.

18. The memory subsystem of claim 14 wherein the register of the memory controller further includes a memory valid field storing a value indicative of the number of functional memory sub-arrays of the plurality of memory sub-arrays.

19. A graphics processing system, comprising:

a bus interface for coupling to a system bus;

a graphics processor coupled to the bus interface to process graphics data;

address and data busses coupled to the graphics processor to transfer address and graphics data to an from the graphics processor;

display logic coupled to the data bus to drive a display;

a memory request bus coupled to the graphics processor to transfer memory and access requests; and

a memory subsystem coupled to the memory request bus to receive and service memory access requests, the memory subsystem comprising:

a first memory array segmented into a plurality of memory sub-arrays, a number of which are functional, each of the functional memory sub-arrays being assigned to a respective block of memory and any faulty memory sub-arrays being left unassigned; and

a memory controller coupled to receive memory access requests to a block of memory to which a functional memory sub-array from the first memory array is assigned and further coupled to the first memory array to access the functional memory sub-array assigned to the requested block of memory.

20. The graphics processing system of claim 19 wherein the first memory array comprises an embedded memory array.

21. The graphics processing system of claim 20 wherein the first memory controller of the memory subsystem includes a register to store a pointer value for each of the functional memory sub-arrays indicative of the block of memory to which a respective functional memory sub-array is assigned.

22. The graphics processing system of claim 19 wherein the memory subsystem further comprises:

a second memory array segmented into a plurality of memory sub-arrays, a number of which are functional, each of the functional memory sub-arrays of the second memory array assigned to a respective block of memory and any faulty memory sub-arrays left unassigned;

a second memory controller coupled to receive memory access requests to a block of memory to which a functional memory sub-array of the second memory array is

assigned and further coupled to the second memory array to access the functional memory sub-array assigned to the requested block of memory; and

a memory bus controller coupled between the first and second memory controllers to pass a memory access request from one memory controller to the other in response to receiving a memory access request to access a memory location within the memory array coupled to the other memory controller.

23. The graphics processing system of claim 20 wherein the first and second memory controllers of the memory subsystem store a value indicative of the number of functional sub-arrays in the first memory array a second memory array, respectively.

24. The graphics processing system of claim 20 wherein the first and second memory controllers of the memory subsystem further store a start and size value for the first and second memory array, respectively, the start and size values defining the addressable memory area of the respective memory array.

25. The graphics processing system of claim 24 wherein the start value stored by the second memory controller of the memory subsystem is the sum of the start value and the size value stored by the first memory controller.

26. A computer system, comprising:

a system processor;

a system bus coupled to the system processor;

a system memory coupled to the system bus; and

a graphics processing system coupled to the system bus, the graphics processing system comprising:

a bus interface for coupling to a system bus;

a graphics processor coupled to the bus interface to process graphics data;

address and data busses coupled to the graphics processor to transfer address and graphics data to an from the graphics processor;

display logic coupled to the data bus to drive a display;

a memory request bus coupled to the graphics processor to transfer memory and access requests; and

a memory subsystem coupled to the memory request bus to receive and service memory access requests, the memory subsystem comprising:

a memory array segmented into a plurality of memory sub-arrays; and

a memory controller coupled to the memory request bus to receive memory access requests and further coupled to access the memory array, the memory controller having a register including a plurality of data fields, the data fields storing a pointer value indicative of which memory sub-arrays are functional and which memory sub-arrays to access in response to the memory controller receiving memory access requests.

27. The computer system of claim 26 wherein the memory array of the graphics processing system comprises an embedded memory array fabricated on a semiconductor substrate with the memory controller.

28. The computer system of claim 26 wherein the memory subsystem of the graphics processing system further comprises:

a second memory array segmented into a plurality of memory sub-arrays;

a second memory controller coupled to access the second memory array and having a register including a plurality of data fields, the data fields of the second memory controller storing a pointer value indicative of which memory sub-arrays of the second memory array are functional and which to access in response to the second memory controller receiving a memory access request; and

a memory controller bus coupled between the memory controller and the second memory controller on which the memory access request may be passed from one memory controller to the other.

29. The computer system of claim 28 wherein the second memory array of the memory subsystem comprises an embedded memory fabricated on the same semiconductor substrate as the memory array.

30. The computer system of claim 26 wherein the register of the memory controller further includes a memory valid field storing a value indicative of the number of functional memory sub-arrays of the plurality of memory sub-arrays.

31. A method of accessing a memory array segmented into a plurality of memory sub-arrays, at least one of the memory sub-arrays being functional, the method comprising:

assigning each functional memory sub-array of the memory array to a respective memory block and leaving any faulty memory sub-arrays unassigned; and

in response to receiving a memory access request to access a particular memory block, accessing the memory sub-array assigned to the particular memory block.

32. The method of claim 31 wherein assigning each functional memory sub-array comprises storing for each memory block a pointer value identifying the respective functional memory sub-array to which it is assigned.

33. The method of claim 32, further comprising storing a valid value indicative of the number of functional memory sub-arrays.

34. The method of claim 31 wherein the memory array comprises an embedded memory array.

35. The method of claim 31, further comprising:

storing start address and size values defining an addressable memory area of the memory array;

determining from the start address and size values whether the particular memory block of the memory access request is assigned to a memory sub-array within the addressable memory area of the memory array; and

servicing the memory access request if the particular memory block is determined to be assigned to a memory sub-array within the addressable memory area of the memory array, otherwise passing the memory access request to another memory controller for servicing.

36. The method of claim 35, further comprising storing second start address and size values defining an addressable memory area of a second memory array, the second start address value equal to the sum of the start address and size values of the addressable memory area of the memory array.

37. A method of accessing an embedded memory array segmented into a plurality of memory sub-arrays, at least one of the memory sub-arrays being functional, the method comprising:

storing for each of a plurality of memory blocks a pointer value identifying a functional memory sub-array assigned thereto;

storing start address and size values defining an addressable memory area of the embedded memory array;

in response to receiving a memory access request to access a particular memory block, determining from the start address and size values whether the particular memory block is assigned to a memory sub-array within the addressable memory area of the embedded memory array; and

accessing the memory sub-array identified by the pointer value stored for the particular memory block if the particular memory block is determined to be assigned to a memory sub-array within the addressable memory area of the embedded memory array, otherwise passing the memory access request to another memory controller for servicing.

38. The method of claim 37, further comprising storing second start address and size values defining an addressable memory area of a second embedded memory array, the second start address value equal to the sum of the start address and size values of the addressable memory area of the embedded memory array.

39. The method of claim 37, further comprising storing a valid value indicative of the number of function memory sub-arrays of the memory array.

APPARATUS AND METHOD FOR DYNAMICALLY DISABLING FAULTY EMBEDDED MEMORY IN A GRAPHIC PROCESSING SYSTEM

ABSTRACT OF THE DISCLOSURE

A memory subsystem and method for ignoring faulty memory sub-arrays and assigning functional memory sub-arrays to accessible memory blocks. The memory subsystem includes a memory array segmented into a plurality of memory sub-arrays and a memory controller that accesses the memory array. The memory controller references a register storing a pointer value that assigns a memory block to each of the functional memory sub-arrays of the memory array. The faulty memory sub-arrays are left unassigned so that they are ignored by the memory controller. Upon receiving a memory access request to a particular memory block, the memory controller accesses the functional memory sub-array assigned to the particular memory block by the pointer value.

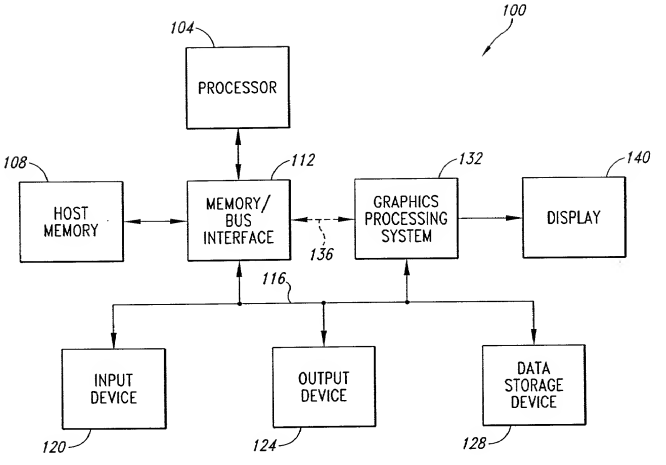
100
104
PROCESSOR
108
HOST MEMORY
112
MEMORY/ BUS INTERFACE
136
132
GRAPHICS PROCESSING SYSTEM
140
DISPLAY
116
INPUT DEVICE
120
OUTPUT DEVICE
124
DATA STORAGE DEVICE
128

Fig. 1

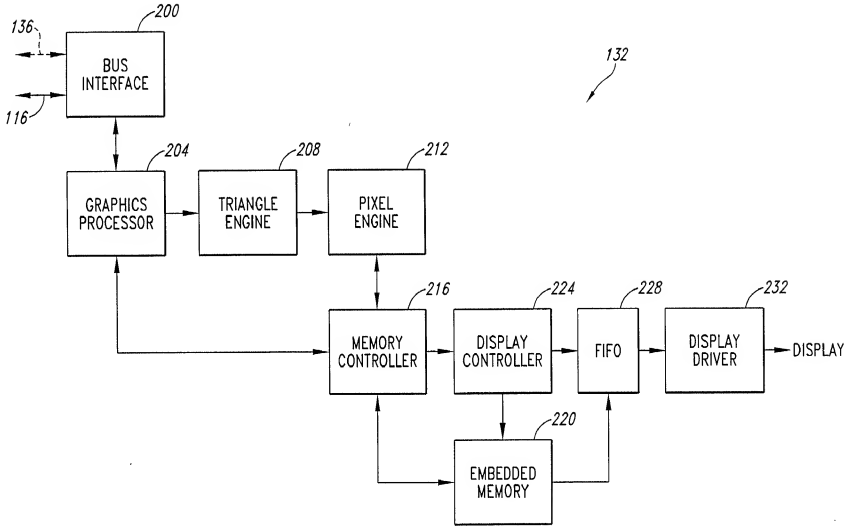
200
136
BUS INTERFACE
116
132
204
208
212
GRAPHICS PROCESSOR
TRIANGLE ENGINE
PIXEL ENGINE
216
224
228
232
MEMORY CONTROLLER
DISPLAY CONTROLLER
FIFO
DISPLAY DRIVER
DISPLAY
220
EMBEDDED MEMORY

Fig. 2

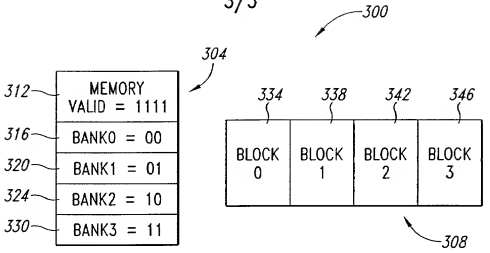

Fig. 3A

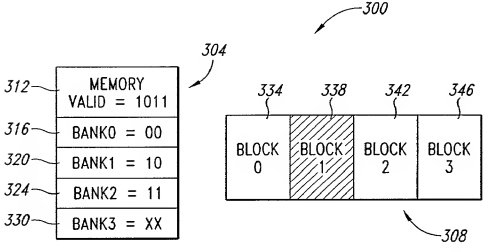

Fig. 3B

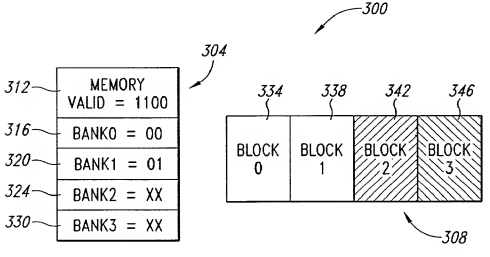

Fig. 3C

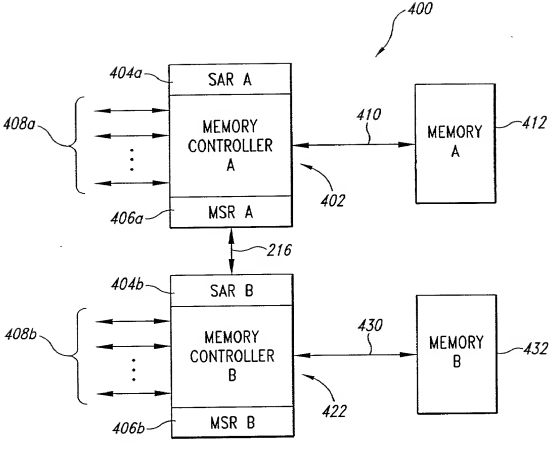
400
404a
SAR A
408a
MEMORY CONTROLLER A
410
MEMORY A
412
402
406a
MSR A
216
404b
SAR B
408b
MEMORY CONTROLLER B
430
MEMORY B
432
422
406b
MSR B

Fig. 4

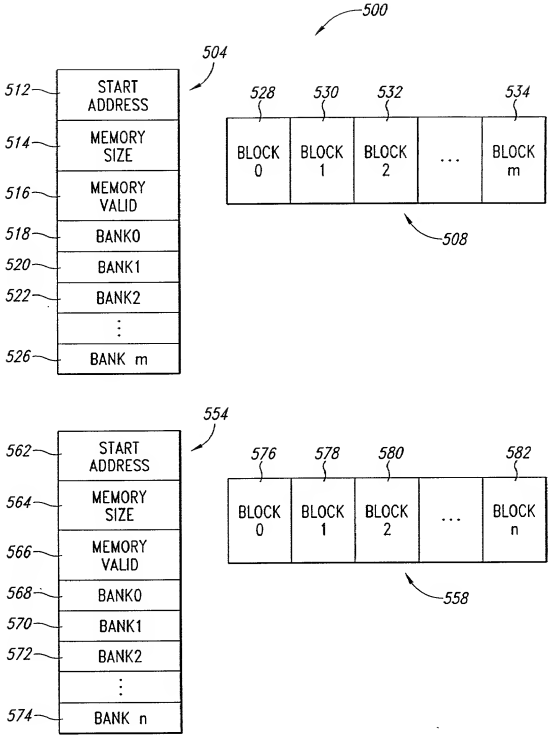
500
504
512
START ADDRESS
514
MEMORY SIZE
516
MEMORY VALID
518
BANK0
520
BANK1
522
BANK2
526
BANK m
528
530
532
534
BLOCK 0
BLOCK 1
BLOCK 2
BLOCK m
508
554
562
START ADDRESS
564
MEMORY SIZE
566
MEMORY VALID
568
BANK0
570
BANK1
572
BANK2
574
BANK n
576
578
580
582
BLOCK 0
BLOCK 1
BLOCK 2
BLOCK n
558

Fig. 5

DECLARATION

As the below-named inventors, we declare that:

Our residences, post office addresses, and citizenships are as stated below under our names.

We believe we are the original, first, and joint inventors of the invention entitled "APPARATUS AND METHOD FOR DYNAMICALLY DISABLING FAULTY EMBEDDED MEMORY IN A GRAPHIC PROCESSING SYSTEM," which is described and claimed in the foregoing specification and for which a patent is sought.

We have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment specifically referred to herein (if any).

We acknowledge our duty to disclose information of which we are aware which is material to patentability and examination of this application in accordance with 37 C.F.R. § 1.56(a).

We further declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further, that these statements were made with the knowledge that the making of willfully false statements and the like is punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and may jeopardize the validity of any patent issuing from this patent application.

James R. Peterson

Date 4/10/2000

Residence	:	City of Portland, County of Washington State of Oregon
Citizenship	:	United States of America
P.O. Address	:	2432 N.W. 141st Place Portland, Oregon 97229

William Radke

William Radke

Date May 11, MM

Residence : City of San Francisco, County of San Francisco
State of California

Citizenship : United States of America

P.O. Address : 4284 26th Street
San Francisco, California 94131

ip \clients\rend-micront\500689.01\decl

EXPRESS MAIL NO. EL476401367US

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants	: James R. Peterson and William Radke	Attorney Docket No.: 500689.01
Filed	: Concurrently herewith	
Title	: APPARATUS AND METHOD FOR DYNAMICALLY DISABLING FAULTY EMBEDDED MEMORY IN A GRAPHIC PROCESSING SYSTEM	

Box Patent Application
Assistant Commissioner for Patents
Washington, DC 20231

ELECTION UNDER 37 C.F.R. §§ 3.71 AND 3.73 AND POWER OF ATTORNEY

Sir:

The undersigned, being Assignee of the entire interest in the above-identified application by virtue of an Assignment filed concurrently herewith, hereby elects, under 37 C.F.R. § 3.71, to prosecute the application to the exclusion of the inventors.

Assignee hereby appoints EDWARD W. BULCHIS, Reg. No. 26,847; JON F. TUTTLE, Reg. No. 25,713; PAUL T. MEIKLEJOHN, Reg. No. 26,569; GLENN P. RICKARDS, Reg. No. 29,428; DALE C. BARR, Reg. No. 40,498; KIMTON N. ENG, Reg. No. 43,605; DAVID E. BOONE, Reg. No. 27,857; SCOTT W. DOYLE, Reg. No. 39,176; REED R. HEIMBECHER, Reg. No. 36,353; JOHN T. KENNEDY, Reg. No. 42,717; GREGORY D. LEIBOLD, Reg. No. 36,408; GARY M. POLUMBUS, Reg. No. 25,364; THOMAS H. YOUNG, Reg. No. 25,796; W. ROBINSON H. CLARK, Reg. No. 41,530; GREGORY J. GLOVER, Reg. No. 34,173; JOHN K. HARROP, Reg. No. 41,817; CHRIS McWHINNEY, Reg. No. 42,875; ALDO NOTO, Reg. No. 35,628; MATTHEW PHILLIPS, Reg. No. 43,403; JOHN W. RYAN, Reg. No. 33,771; AMI P. SHAH, Reg. No. 42,143; SEAN S. WOODEN, Reg. No. 43,997; MICHAEL C. GILCHRIST, Reg. No. 40,619; BRIAN J. LAURENZO, Reg. No. 34,207; SHANE COLEMAN, Reg. No. 44,623; RONALD J. BROWN, Reg. No. 29,016; DAVID E. BRUHN, Reg. No. 36,762; DAVID N. FRONEK, Reg. No. 25,678; JOSEPH F. HAAG, Reg. No. 42,612; STUART R. HEMPHILL, Reg. No. 28,084; GRANT A.

JOHNSON, Reg. No. 42,696; KENNETH E. LEVITT, Reg. No. 39,747; NIALL A. MACLEOD, Reg. No. 41,963; SCOTT A. MARKS, Reg. No. 44,902; DEVAN V. PADMANABHAN, Reg. No. 38,262; GERALD H. SULLIVAN, Reg. No. 37,243; BRIAN PARK, Reg. No. 45,519; MARK W. ROBERTS, Reg No. 46,160; STEVEN H. ARTERBERRY, Reg. No. 46,314; PAUL F. RUSYN, Reg. No. 42,118; of the firm of DORSEY & WHITNEY LLP, along with MICHAEL L. LYNCH, Reg. No. 30,871; LIA M. PAPPAS, Reg. No. 34,095; WALTER D. FIELDS, Reg. No. 37,130; CHARLES B. BRANTLEY, II, Reg. No. 38,086; KEVIN D. MARTIN, Reg. No. 37,882; and DAVID J. PAUL, Reg. No. 34,692, of MICRON TECHNOLOGY, INC., 8000 South Federal Way, Boise, Idaho 83706-9632, as its attorneys to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. Please direct all telephone calls to **Edward W. Bulchis** at (206) 903-8800 and telecopies to (206) 903-8820.

Please direct all communications to:

Edward W. Bulchis, Esq.
Dorsey & Whitney LLP
1420 Fifth Avenue, Suite 3400
Seattle, Washington 98101
(206) 903-8800

Pursuant to 37 C.F.R. § 3.73, the undersigned duly authorized designee of Assignee certifies that the evidentiary documents have been reviewed, specifically the Assignment to MICRON TECHNOLOGY, INC., filed concurrently herewith for recording, a copy of which is attached hereto, and certifies that to the best of my knowledge and belief, title remains in the name of the Assignee.

MICRON TECHNOLOGY, INC.
ASSIGNEE

6-1-00
DATE

Michael L. Lynch
Chief Patent Counsel

Enclosure:
Copy of Assignment

rendition\500689.01\500689.01 election and poa.doc